

ABSTRACT

A method and system for providing "just-in-time" dynamic loading and unloading of libraries. Library code is loaded into memory just prior to the actual execution of a routine in that library and is unloaded from memory after the execution of the library routine is completed. According to the present invention, a library structure having a library loader and a library implementation module. The library loader provides the same entry points as the library itself and contains code to drive the loading and unloading of the library implementation module which actually contains the code to implement the library call. At the beginning of the execution of an application or other software module, the library loaders for libraries needed by that application or other software module, are loaded into memory and unloaded upon completion of that application or other software module. The library loaders control the loading and unloading of the library implementation modules, which contain the actual code which implements the library call, so that the library code itself is only in memory while in use during execution of the library call. A sequence of library calls to the same library is preferably detected and managed so as to avoid unnecessary unloading/reloading of the same library.